# Module 2 – Set Operators

## Overview

The main purpose of this lab is to familiarise yourself with key set operators with TSQL.

- Union
- Intersect
- Except
- Apply

## Objectives

At the end of this lab, you will be able to:

- Choose the correct set operators for your needs
- Employ each of the set operators

## Setup: Launch SQL Server Management Studio

1. On the Start menu, click All Programs, click Microsoft SQL Server 2014 and then click SQL Server Management Studio.
2. The Microsoft SQL Server Management Studio window opens, and then the Connect to Server dialog box will appear.
3. In the Connect to Server dialog box, click Connect to accept the default settings.
4. On the toolbar, click New Query, and either select the QATSQLPLUS database in the Available Databases box, or type USE QATSQLPLUS in the query window.
5. If you wish, you may save your queries to My Documents or the desktop. All modules are separate and you will not require any queries from this module in any later module.

## Exercise 1: Contacts list

In this exercise, you will create a query to return a contacts list of both vendors and trainers.

The main tasks for this exercise are as follows:

1. Create a query to return the contact name and phone number for each vendor
2. Create a query to return the trainer name and phone number for each trainer
3. Combine the previous two queries into a single result set

### Task 1: Vendor contacts

1. Write a query to return the Vendor contact information. The columns returned should be:
    - VendorName
    - PhoneNumber
2. Test the query.
3. Keep the query window open for the following tasks.

### Task 2: Trainer contacts

1. Write a query to return the Trainer contact information. The columns returned should be:
    - TrainerName
    - PhoneNumber
2. Test the query.
3. Keep the query window open for the following tasks.

### Task 3: Complete contact list

1. Using the queries from Tasks 1 and 2, create a query to combine the two data sets.
2. Test the query.
    - How many rows were returned?
    - Are there duplicate rows returned?
3. Alter the query to show an extra column what show either “Vendor” or “Trainer” for each row. The new column should be called ContactType.
4. Test the query.
5. Keep the query window open for the following tasks.

## Exercise 2: Course Overlap

In this exercise, you find all delegates who have attended both the QATSQL and QATSQLPLUS courses.

The main tasks for this exercise are as follows:

1. Create a query that returns the list of DelegateID who have attended the QATSQL
2. Create a query that returns the list of DelegateID who have attended the QATSQLPLUS
3. Find the list of DelegateID which appear in both of the original queries

### Task 1 : Create the QATSQL delegates query

1. Write a query to join the DelegateAttendance, CourseRun and Course tables. The only columns returned should be DelegateID where the CourseName = QATSQL.
2. Test the query. This should return 11 rows.
3. Keep the query window open for the following tasks.

### Task 2 : Create the QATSQLPLUS delegates query

1. Write a query to join the DelegateAttendance, CourseRun and Course tables. The only columns returned should be DelegateID where the CourseName = QATSQLPLUS.
2. Test the query. This should return 7 rows.
3. Keep the query window open for the following tasks.

### Task 3: Create a query showing the overlap

1. Using the queries from tasks 1 and 2, create query to return the list DelegateID appearing in both sets.
2. Test the query. This should return 6 rows.
3. Keep the query window open for the following tasks.

## Exercise 3: Course Disjoint

In this exercise, you find all delegates who have attended the QATSQL course but have not attended the QATSQLPLUS courses.

The main tasks for this exercise are as follows:

1. Find the list of DelegateID which have attended QATSQL but not QATSQLPLUS.
2. Find the list of DelegateID which have attended QATSQLPLUS but not QATSQL.

### Task 1: Create a query showing the disjoint

1. Using the query from exercise 2 task 3, update the query to return the list DelegateID attending QATSQL but not QATSQLPLUS.
2. Test the query. This should return 4 rows.
3. Update the query to return the list DelegateID attending QATSQLPLUS but not QATSQL.
4. Test the query. This should return 1 row.
5. Keep the query window open for the following tasks.

## Answers

The answers below are for example only. Coding style and order of columns should not matter.

**Ex 1**

```
--Task 1 :
SELECT ContactName, PhoneNumber
   FROM dbo.Vendor

--Task 2 :
SELECT TrainerName, PhoneNumber
   FROM dbo.Trainer

--Task 3 (1a) :
SELECT ContactName, PhoneNumber
   FROM dbo.Vendor
UNION
SELECT TrainerName, PhoneNumber
   FROM dbo.Trainer

--Task 3 (1b) :
SELECT ContactName, PhoneNumber
   FROM dbo.Vendor
UNION ALL
SELECT TrainerName, PhoneNumber
   FROM dbo.Trainer

--Task 3 (3) :
SELECT 'Vendor' AS ContactType,ContactName, PhoneNumber
   FROM dbo.Vendor
UNION
SELECT 'Trainer' AS ContactType,TrainerName, PhoneNumber
   FROM dbo.Trainer
```

**Ex 2**

```
USE QATSQLPLUS
GO

--Task 1:
SELECT DelegateID
   FROM dbo.DelegateAttendance AS DA
      INNER JOIN dbo.CourseRun AS CR
         ON DA.CourseRunID = CR.CourseRunID
      INNER JOIN dbo.Course AS C
         ON C.CourseID = CR.CourseID
   WHERE C.CourseName = 'QATSQL'
--Task 2:
SELECT DelegateID
   FROM dbo.DelegateAttendance AS DA
      INNER JOIN dbo.CourseRun AS CR
         ON DA.CourseRunID = CR.CourseRunID
      INNER JOIN dbo.Course AS C
         ON C.CourseID = CR.CourseID
   WHERE C.CourseName = 'QATSQLPLUS'

--Task 3:
SELECT DelegateID
   FROM dbo.DelegateAttendance AS DA
      INNER JOIN dbo.CourseRun AS CR
         ON DA.CourseRunID = CR.CourseRunID
      INNER JOIN dbo.Course AS C
         ON C.CourseID = CR.CourseID
   WHERE C.CourseName = 'QATSQL'
INTERSECT
SELECT DelegateID
   FROM dbo.DelegateAttendance AS DA
      INNER JOIN dbo.CourseRun AS CR
         ON DA.CourseRunID = CR.CourseRunID
      INNER JOIN dbo.Course AS C
         ON C.CourseID = CR.CourseID
   WHERE C.CourseName = 'QATSQLPLUS'
GO
```

**Ex 3**

```
--Task 1:
SELECT DelegateID
   FROM dbo.DelegateAttendance AS DA
      INNER JOIN dbo.CourseRun AS CR
         ON DA.CourseRunID = CR.CourseRunID
      INNER JOIN dbo.Course AS C
         ON C.CourseID = CR.CourseID
   WHERE C.CourseName = 'QATSQL'
EXCEPT
SELECT DelegateID
   FROM dbo.DelegateAttendance AS DA
      INNER JOIN dbo.CourseRun AS CR
         ON DA.CourseRunID = CR.CourseRunID
      INNER JOIN dbo.Course AS C
         ON C.CourseID = CR.CourseID
   WHERE C.CourseName = 'QATSQLPLUS'
--Task 2:
SELECT DelegateID
   FROM dbo.DelegateAttendance AS DA
      INNER JOIN dbo.CourseRun AS CR
         ON DA.CourseRunID = CR.CourseRunID
      INNER JOIN dbo.Course AS C
         ON C.CourseID = CR.CourseID
   WHERE C.CourseName = 'QATSQLPLUS'
EXCEPT
SELECT DelegateID
   FROM dbo.DelegateAttendance AS DA
      INNER JOIN dbo.CourseRun AS CR
         ON DA.CourseRunID = CR.CourseRunID
      INNER JOIN dbo.Course AS C
         ON C.CourseID = CR.CourseID
   WHERE C.CourseName = 'QATSQL'
```